Course title: TESTING AND QUALITY OF SOFTWARE PRODUCTS

Lecturers	Full Prof. Valentina Kirinić, Ph.D. Marko Mijač, Ph.D.
Language of instruction	Croatian and English
Study level	Bachelor
Study programme	Information and Business Systems
Semester	6th (summer)
ECTS	6
Goal	The goal of the course is to enable students to participate in planning, conducting and documenting software product testing, as well as performing quality evaluation activities. Through lectures and laboratory exercises, students will be familiarized with different principles, best practices and commonly used techniques and tools for testing, debugging and profiling software products. This is complemented with widely accepted metrics and techniques for software quality evaluation. Demonstrated knowledge will serve as a preparation for students' own projects.
General and specific learning outcomes	
Content	**1. An introduction to testing and quality of software products** Reflection on your own competences, experiences and practices in the field of software product testing and quality. Importance and good practice in the field of software product testing and quality. Motivation: professions / jobs related to software product testing and quality. Definition and key concepts. Validation, verification, testing and quality assurance of software products. **2. The fundamentals of software product quality** Quality within a software product development project. Planning, assuring, controlling and improving software product quality. Software product quality standards and recommendations. Software quality management techniques (static and dynamic). Measures / metrics, models and tools for software product quality assurance. Documenting the process of software product quality evaluation. **3. The fundamentals of software product testing** Definition and key concepts in software testing. Testing goals. Position and the role of testing activities in the software process. Similarities and differences with regard to other related activities (SQM, formal verification, debugging, software implementation). Software errors - causes and effects. Testing principles. **4. Software product testing process and its documentation** Planning, preparation, design, organization and execution of tests. **5. Phases in software product testing** Development testing. Release testing. User testing. **6. Software product testing techniques** Categorization of testing techniques. White-box and Black-box techniques. Experience-based and intuition-based techniques. Source-code based techniques. Error-based techniques… **7. Agile testing and test-driven software development**

	Fundamental principles of test-driven development. Fundamental and advanced concepts in unit testing. Patterns and good practices. **8. Testing non-functional requirements** Testing performance, reliability, security of software products. **9. Tools, libraries, and software frameworks for testing** Benefits of automated testing. Examples of popular tools. Integration of testing into a process of automated delivery of software products (DevOps).
Exercises	Laboratory exercises are in line with the content of the lectures and serve as a preparation for the student's own project. Laboratory exercises will make use of appropriate tools for software product testing and quality evaluation. Appropriate tools for software product testing and quality evaluation will be used in the laboratory exercises.
Realization and examination	Classes: lectures and exercises Exam: theoretical knowledge is evaluated during the semester through written preliminary exams (or written and oral final exam). Students demonstrate acquired practical skills by working on a project and defending it.
Related courses	1. Software quality and testing (University of Gothenburg, Sweden) 2. Software testing (Johannes Kepler University Linz, Austria) 3. Testing and quality (University of Ljubljana, Faculty of Computer and Information Science, Slovenia) 4. Software quality and testing (University of Reading, UK)
Literature	Basic: Lewis, W. E. (2016). Software testing and continuous quality improvement. Auerbach publications. Desai, S., & Srivastava, A. (2016). Software testing: A practical approach. PHI Learning Pvt. Ltd. Fenton, N., & Bieman, J. (2014). Software metrics: a rigorous and practical approach. CRC press. Additional: ISO/IEC/IEEE 29119-1:2013 Software and systems engineering -- Software testing -- Part 1: Concepts and definitions ISO/IEC/IEEE 29119-2:2013 Software and systems engineering -- Software testing -- Part 2: Test processes ISO/IEC/IEEE 29119-3:2013 Software and systems engineering -- Software testing -- Part 3: Test documentation ISO/IEC/IEEE 29119-4:2015 Software and systems engineering -- Software testing -- Part 4: Test techniques ISO/IEC 30130:2016 Software engineering -- Capabilities of software testing tools Beck, Kent (2002). Test-driven development, by example. Addison Wesley Osherove, Roy (2009). The art of Unit testing. Manning Publications